AMENDMENTS TO THE CLAIMS:

This listing of the claims will replace all prior versions, and listings, of the claims in this application.

Listing of Claims:

1. (Currently amended) A computer program product stored on a computer readable storage medium, comprising computer readable program code instructions for retaining data in a data cache comprising virtual tracks of data by performing:

referencing dirty data stored in the data cache using a first least recently used list, said first least recently used list having a head and a tail; and

referencing clean data stored in the data cache using a second least recently used list, said second least recently used list having a head and a tail;

wherein the dirty data is destaged from the data cache when the dirty data reaches the tail of the first least recently used list and the clean data is purged from the data cache when the clean data reaches the tail of the second least recently used list.

2. (Currently Amended) A computer program product as claimed in claim 1, wherein the dirty data is destaged to a data storage device and deleted from the first least recently used list and reference to the ~~destaged~~ dirty data so destaged is added to the second least recently used list and a copy of the ~~destaged~~ dirty data so destaged is retained in the cache as clean data.

3. (Previously Presented) A computer program product as claimed in claim 1, wherein a read command comprising a cache miss fetches data from a data storage device and the data is retained in the cache with a reference in the second least recently used list.

4. (Previously Presented) A computer program product as claimed in claim 1, further performing: keeping a flag with each data reference in the first least recently used list indicating whether or not the data has been read while on the first least recently used list.

5. (Previously Presented) A computer program product as claimed in claim 1, wherein, if the data was read when referenced in the first least recently used list, the data is added to the head of the second least recently used list when the data is destaged.

6. (Previously Presented) A computer program product as claimed in claim 1, wherein, if the data was not read when referenced in the first least recently used list, the data is one of maintained in the current position in the second least recently used list or discarded.

7. (Previously Presented) A computer program product as claimed in claim 4, wherein keeping the flag comprises including a timestamp each time the data is read.

8. (Previously Presented) A computer program product as claimed in claim 1, wherein a virtual track of the data comprises partially dirty data and partially clean data and the virtual track is referenced in both the first least recently used list and second least recently used list.

9. (Currently Amended) A data storage system comprising:

a storage controller comprising a cache;

wherein the cache comprises a first least recently used list for referencing dirty data which is stored in the cache, said first least recently used list having a head and a tail, and a second least recently used list for referencing clean data which is stored in the cache, said second least recently used list having a head and a tail;

wherein the dirty data is destaged from the cache when the dirty data reaches the tail of the first least recently used list and the clean data is purged from the cache when the clean data ~~it~~ reaches the tail of the second least recently used list.

10. (Previously Presented) A data storage system as claimed in claim 9, wherein the dirty data is destaged to ~~a~~ the data storage device and deleted from the first least recently used list and reference to the destaged data is added to the second least recently used list and a copy of the destaged data is retained in the cache as clean data.

11. (Previously Presented) A data storage system as claimed in claim 9, wherein a read command comprising a cache miss fetches data from the data storage device and the data is retained in the cache with a reference in the second least recently used list.

12. (Previously Presented) A data storage system as claimed in claim 9, wherein a flag is provided with each data reference in the first least recently used list indicating whether or not the data has been read while on the first least recently used list.

13. (Previously Presented) A data storage system as claimed in claim 9, wherein, if the data was read when referenced in the first least recently used list, the data is added to the head of the second least recently used list when the data is destaged.

14. (Previously Presented) A data storage system as claimed in claim 9, wherein, if the data was not read when referenced in the first least recently used list, the data is one of maintained in the current position in the second least recently used list or discarded.

15. (Previously Presented) A data storage system as claimed in claim 12, wherein the flag comprises a timestamp each time the data is read and the timestamp is adapted for prioritizing the position of the data reference in the second least recently used list.

16. (Previously Presented) A data storage system as claimed in claim 9, wherein a region of the data comprises partially dirty data and partially clean data and the region is referenced in both the first least recently used list and second least recently used list.

17. (Currently Amended) A method for data retention in a data cache, comprising:

referencing dirty data stored in the data cache in a first least recently used list, said first least recently used list having a head and a tail; and

referencing clean data in the data cache in a second least recently used list, said second least recently used list having a head and a tail;

wherein the dirty data is destaged from the data cache when the dirty data reaches the tail of the first least recently used list and the clean data is purged from the data cache when the clean data reaches the tail of the second least recently used list.

18. (Original) A computer program product as in claim 1, wherein the dirty data comprises data received from a host computer.

19. (Original) A computer program product as in claim 1, wherein the clean data comprises data destaged to a storage device.

20. (Original) A computer program product as in claim 7, wherein the timestamp is adapted for prioritizing the position of the data reference in the second least recently used list.

21. (Original) A computer program product as in claim 1, wherein when a virtual track comprising dirty data is inserted into the data cache, a reference to the virtual track is added to the head of the first least recently used list.

22. (Original) A computer program product as in claim 1, wherein when a virtual track comprising clean data is inserted into the data cache, a reference to the virtual track is added to the head of the second least recently used list.

23. (Original) A computer program product as in claim 1, wherein when dirty data is merged into a virtual track referenced in the first least recently used list, a reference to the virtual track is moved to the head of the first least recently used list.

24. (Original) A computer program product as in claim 1, wherein when dirty data is merged into a virtual track referenced in the second least recently used list, a reference to the virtual track is added to the head of the first least recently used list and the reference to the virtual track remains in the second least recently used list if the virtual track comprises any clean pages.

25. (Original) A computer program product as in claim 1, wherein when dirty data is merged into a virtual track referenced in both the first least recently used list and the second least recently used list, the reference in the first least recently used list is moved to the head of the first least recently used list and the reference in the second least recently used list remains if the virtual track comprises any clean pages.

26. (Original) A computer program product as in claim 1, wherein when clean data is merged into a virtual track referenced in the first least recently used list a read flag is set for the virtual track; the reference is left at the current location in the first least recently used list and a reference to the virtual track is added to the head of the second least recently used list if the virtual track comprises clean pages.

27. (Original) A computer program product as in claim 1, wherein when clean data is merged into a virtual track, a reference to the virtual track in the second least recently used list is moved to the head of the second least recently used list.

28. (Original) A computer program product as in claim 1, wherein when clean data is merged into a virtual track referenced in both the first least recently used list and the second least recently used list, a read flag is cleared for the virtual track, and the reference to the virtual track in the first least recently used list is left at the current location and the reference to the virtual track in the second least recently used list is moved to the head of the second least recently used list.

29. (Original) A computer program product as in claim 1, wherein when a last clean page of data is purged from a virtual track referenced at the end of the second least recently used list, the reference to the virtual track is deleted from the end of the second least recently used list..

30. (Original) A computer program product as in claim 1, wherein when a virtual track referenced in the first least recently used list no longer contains any dirty data, the reference to the virtual track is deleted.

31. (Original) A computer program product as in claim 1, wherein when a virtual track referenced in the second least recently used list no longer contains any clean pages, the reference to the virtual track is deleted.

32. (Original) A data storage system as in claim 9, wherein when the data is destaged, the data is written to an external storage device.

33. (Original) A data storage system as in claim 9, further comprising a data storage device coupled to the storage controller.

34. (Original) A data storage system as in claim 33, wherein the data storage device comprises at least one disk drive.

35. (Currently Amended) A host computer connected to a data storage system comprising:

a storage controller comprising a cache;

wherein the cache comprises a first least recently used list for referencing dirty data which is stored in the cache, said first least recently used list having a head and a tail, and a second least recently used list for referencing clean data which is stored in the cache, said second least recently used list having a head and a tail;

wherein the dirty data is destaged from the cache when the dirty data reaches the tail of the first least recently used list and the clean data is purged from the cache when the clean data reaches the tail of the second least recently used list.

36. (Currently Amended) A data storage system adapted for retaining data in a data cache, the system comprising:

means for referencing dirty data stored in the data cache using a first least recently used list, said first least recently used list having a head and a tail;

means for referencing clean data stored in the data cache using a second least recently used list, said second least recently used list having a head and a tail; and,

means for destaging the dirty data from the data cache when the dirty data reaches the tail of the first least recently used list and means for purging the clean data from the data cache when the clean data reaches the tail of the second least recently used list.

37. (Original) A computer program stored on a computer readable medium and executable by a data processor of a data storage unit that is coupled to a plurality of disk drives, comprising program code, responsive to a data write being received in a data cache, for placing a data descriptor for the data at a head of a Least Recently Written (LRW) list as dirty data, and for maintaining a flag with the data descriptor for indicating if the data is read; said program code being further responsive to the data descriptor moving down the LRW list, until it reaches a tail of the LRW list, for destaging the data to at least some of said plurality of disk drives and for determining if the data descriptor is currently in a Least Recently Read (LRR) list and, if the data descriptor is currently in the LRR list, maintaining the data descriptor at its current location in the LRR list, while if the data descriptor is currently not in the LRR list, testing the flag to determine if the data has been read while the data descriptor was in the LRW list and, if the data was read while in the LRW list, the data descriptor is placed at the head of the LRR list, otherwise the data is discarded.

38. (Original) A computer program as in claim 37, where said plurality of disk drives comprise a RAID 5 array of disk drives.

39. (Original) A computer program as in claim 37, where said flag comprises a timestamp.